



Name, Surname: Can Yılmaz ALTINIĞNE

Student Number: 150130132

REPORT

Firstly, I created two structs for cards. One of them includes next pointer which is used in queues and stacks and the other one is for unshuffled deck which is an array. I created an array for unshuffled deck which has variables of card struct not having next pointers. A deck swapper function and recursive shuffle function is created. This recursive shuffle function is randomly changes two cards' place in the unshuffled array. This function calls itself 200 times recursively. Then a recursive assign function is created and this function is assigning the cards from array to shuffled deck queue. It calls itself 52 times. In order to call recursive functions I used global variables. After all, I created the most important function in the program which is recursiveThirteenth(). This function dequeues the card from shuffled deck while there are cards in it. The function gets an int value which is the order of counting. This function looks at the cards order (FIFO) if it equals to value of the function takes which is the order of counting, the function enqueues the card to special space queue and it writes the card and “match” to text file and if the card is one of the A, K, Q, J function increases the global sum value by 10. Else it increases by the value of the card and it makes global variable and temporary space is cleaned (LIFO) and the cards from it are enqueued to shuffled deck. In the end the function takes 1 and starts counting another 13s. If the cards value is not equal to value that function gets which is order of counting, the card is pushed to temporary space and card and “unmatch” is

written in text file and fortunesOrder variable which is the global value that the function gets increases by 1. If this variable is equal to 14, it means that 13s counting is made but no card is found in that 13s. So the temporary space is cleaned this time but the cards in it are not being enqueued to shuffled deck. Else our function calls itself with fortunesOrder value recursively. In main function these functions are called and shuffled deck, fortune-telling steps and result (global sum variable) are written to a text file.

```
Masaüstü — -bash — 80x24
Can-MacBook-Pro:desktop canaltinigne$ g++ odev3.cpp -o deneme
Can-MacBook-Pro:desktop canaltinigne$ ./deneme
Output file has been created. Program exits.....
Can-MacBook-Pro:desktop canaltinigne$
```

```
output.txt ~
Shuffled Deck:
1:D:J
2:H:10
3:D:5
4:C:10
5:S:7
6:D:Q
7:S:J
8:C:5
9:H:5
10:C:K
11:D:7
12:S:A
13:D:A
14:D:6
15:D:2
16:D:K
17:H:A
18:H:6
19:S:6
20:C:10
21:H:9
22:D:3
23:H:J
24:D:9
25:H:7
26:D:8
27:C:9
28:S:5
29:H:K
30:S:K
31:S:3
32:S:Q
33:S:4
34:C:4
35:C:2
36:D:10
37:D:4
38:C:7
39:S:9
40:C:A
41:H:3
42:H:Q
43:C:8
44:C:6
45:S:2
46:C:J
47:C:3
48:H:2
49:H:8
50:S:8
51:S:10
52:H:4
Exiting fortune telling steps...
```

Fortune-telling steps:

1:0:J:unmatch
2:H:10:unmatch
3:0:5:unmatch
4:C:0:unmatch
5:5:7:unmatch
6:0:0:unmatch
7:5:J:unmatch
8:C:5:unmatch
9:H:5:unmatch
10:C:K:unmatch
11:0:7:unmatch
12:5:A:unmatch
13:D:A:unmatch
1:0:6:unmatch
2:0:2:match
1:0:K:unmatch
2:H:A:unmatch
3:H:6:unmatch
4:5:6:unmatch
5:C:10:unmatch
6:H:9:unmatch
7:0:3:unmatch
8:H:J:unmatch
9:0:9:match
1:H:7:unmatch
2:0:8:unmatch
3:C:9:unmatch
4:5:5:unmatch
5:H:K:unmatch
6:5:K:unmatch
7:5:3:unmatch
8:5:0:unmatch
9:5:4:unmatch
10:C:4:unmatch
11:C:2:unmatch
12:D:10:unmatch
13:D:4:unmatch
1:C:7:unmatch
2:5:9:unmatch
3:C:A:unmatch
4:H:3:unmatch
5:H:0:unmatch
6:C:8:unmatch
7:C:6:unmatch
8:5:2:unmatch
9:C:J:unmatch
10:C:3:unmatch
11:H:2:unmatch
12:H:8:unmatch
13:5:8:unmatch
1:5:10:unmatch
2:H:4:unmatch
3:0:6:unmatch

4:H:J:unmatch
5:0:3:unmatch
6:H:9:unmatch
7:C:10:unmatch
8:5:6:unmatch
9:H:6:unmatch
10:H:A:unmatch
11:D:K:unmatch

Fortune Results:

D 2
D 9

Sum: 11

Bad luck ! Your wish won't come true !